

IN THE CLAIMS

1. (Currently Amended) A method for designing a computer program, comprising:

accessing a plurality of domain ~~rules for a military theory~~, rules, each domain rule being ~~invariant~~, the plurality of domain rules comprising a plurality of military theory domain rules ~~setting an objective to destroy an enemy's combat forces~~; invariant;

defining a domain from the domain rules, the domain used to determine a problem space and a solution space;

identifying one or more requirements of the domain from one or more supplemental sources;

generating a model that establishes the requirements of the domain;

displaying accessing a plurality of business ~~rules for the military theory~~, rules, each business rule being ~~variable~~, the plurality of business rules comprising a plurality of rules of engagement; variable;

~~selecting one or more rules of engagement in response to a user selection~~;

~~customizing the one or more rules of engagement~~;

associating the one or more business rules ~~of engagement~~ with a ~~procedure~~; the model;

~~associating the military theory domain rules with the procedure~~;

~~displaying a model representing the procedure~~; and

generating a code corresponding to the ~~procedure~~ model in order to design a computer program.

2. (Original) The method of Claim 1, further comprising:

collecting the domain rules and the business rules;

allocating the domain rules and the business rules to a plurality of use cases;

realizing the use cases; and

assessing the domain rules and the business rules in accordance with the realization.

3. (Original) The method of Claim 1, further comprising:
checking a syntax of the code; and
providing a notification if a syntax error is detected.
4. (Original) The method of Claim 1, further comprising:
checking a logical consistency of the code; and
providing a notification if a logical inconsistency is detected.
5. (Original) The method of Claim 1, further comprising:
checking a compatibility between the model and the code; and
providing a notification if an inconsistency is detected.
6. (Previously Presented) The method of Claim 1, wherein the model is
expressed according to a modeling language.

7. (Currently Amended) Logic for designing a computer program, the logic embodied in a computer-readable medium and when executed by a computer operable to:

~~access a plurality of domain rules for a military theory; rules, each domain rule being invariant, the plurality of domain rules comprising a plurality of military theory domain rules setting an objective to destroy an enemy's combat forces; invariant;~~

define a domain from the domain rules, the domain used to determine a problem space and a solution space;

identify one or more requirements of the domain from one or more supplemental sources;

generate a model that establishes the requirements of the domain;

~~display access a plurality of business rules for the military theory; rules, each business rule being variable, the plurality of business rules comprising a plurality of rules of engagement; variable;~~

~~select one or more rules of engagement in response to a user selection;~~

~~customize the one or more rules of engagement;~~

~~associate the one or more business rules of engagement with a procedure; the model;~~

~~associate the military theory domain rules with the procedure;~~

~~display a model representing the procedure; and~~

~~generate a code corresponding to the procedure model in order to design a computer program.~~

8. (Original) The logic of Claim 7, further operable to:

collect the domain rules and the business rules;

allocate the domain rules and the business rules to a plurality of use cases;

realize the use cases; and

assess the domain rules and the business rules in accordance with the realization.

9. (Original) The logic of Claim 7, further operable to:

check a syntax of the code; and

provide a notification if a syntax error is detected.

10. (Original) The logic of Claim 7, further operable to:
check a logical consistency of the code; and
provide a notification if a logical inconsistency is detected.

11. (Original) The logic of Claim 7, further operable to:
check a compatibility between the model and the code; and
provide a notification if an inconsistency is detected.

12. (Previously Presented) The logic of Claim 7, wherein the model is expressed
according to a modeling language.

13. (Currently Amended) A system for designing a computer program, comprising:

a database operable to store a plurality of domain ~~rules for a military theory, rules,~~ each domain rule being ~~invariant, the plurality of domain rules comprising a plurality of military theory domain rules setting an objective to destroy an enemy's combat forces;~~ invariant; and

a server coupled to the database and operable to:

define a domain from the domain rules, the domain used to determine a problem space and a solution space;

identify one or more requirements of the domain from one or more supplemental sources;

generate a model that establishes the requirements of the domain;

display access a plurality of business ~~rules for the military theory, rules,~~ each business rule being ~~variable, the plurality of business rules comprising a plurality of rules of engagement;~~ variable;

~~select one or more rules of engagement in response to a user selection;~~

~~customize the one or more rules of engagement;~~

associate the one or more business ~~rules of engagement~~ with a ~~procedure;~~ the model;

~~associate the military theory domain rules with the procedure;~~

~~display a model representing the procedure;~~ and

generate a code corresponding to the ~~procedure~~ model in order to design a computer program.

14. (Original) The system of Claim 13, the server further operable to:

collect the domain rules and the business rules;

allocate the domain rules and the business rules to a plurality of use cases;

realize the use cases; and

assess the domain rules and the business rules in accordance with the realization.

15. (Original) The system of Claim 13, the server further operable to:
check a syntax of the code; and
provide a notification if a syntax error is detected.

16. (Original) The system of Claim 13, the server further operable to:
check a logical consistency of the code; and
provide a notification if a logical inconsistency is detected.

17. (Original) The system of Claim 13, the server further operable to:
check a compatibility between the model and the code; and
provide a notification if an inconsistency is detected.

18. (Previously Presented) The system of Claim 13, wherein the model is
expressed according to a modeling language.

19. (Currently Amended) A system for designing a computer program, comprising:

means for accessing a plurality of domain ~~rules for a military theory~~; rules, each domain rule being ~~invariant~~, ~~the plurality of domain rules comprising a plurality of military theory domain rules setting an objective to destroy an enemy's combat forces~~; invariant;

means for defining a domain from the domain rules, the domain used to determine a problem space and a solution space;

means for identifying one or more requirements of the domain from one or more supplemental sources;

means for generating a model that establishes the requirements of the domain;

means for ~~displaying~~ accessing a plurality of business ~~rules for the military theory~~; rules, each business rule being ~~variable~~, ~~the plurality of business rules comprising a plurality of rules of engagement~~; variable;

~~means for selecting one or more rules of engagement in response to a user selection~~;

~~means for customizing the one or more rules of engagement~~;

means for associating the one or more business rules of engagement with a ~~procedure~~; the model;

~~means for associating the military theory domain rules with the procedure~~;

~~means for displaying a model representing the procedure~~; and

means for generating a code corresponding to the ~~procedure~~ model in order to design a computer program.

20. (Currently Amended) A method for designing a computer program, comprising:

collecting a plurality of domain rules for a military theory, allocating the domain rules to a plurality of use cases, realizing the use cases, assessing the domain rules in accordance with the realization, and accessing the domain rules, each domain rule being invariant, the plurality of domain rules comprising a plurality of military theory domain rules setting an objective to destroy an enemy's combat forces;

defining a domain from the domain rules, the domain used to determine a problem space and a solution space;

identifying one or more requirements of the domain from one or more supplemental sources;

generating a model that establishes the requirements of the domain;

displaying a plurality of business rules for the military theory, each business rule being variable, the plurality of business rules comprising a plurality of rules of engagement;

selecting one or more rules of engagement in response to a user selection;

customizing the one or more rules of engagement;

associating the one or more rules of engagement with ~~a procedure~~; the model;

associating the military theory domain rules with the ~~procedure~~ model

~~displaying a model representing the procedure, the model expressed according to a modeling language;~~

generating a code corresponding to the ~~procedure~~ model in order to design a computer program;

checking a syntax of the code, and providing a notification if a syntax error is detected;

checking a logical consistency of the code, and providing a notification if a logical inconsistency is detected; and

checking a compatibility between the model and the code, and providing a notification if an inconsistency is detected.

21. (Currently Amended) A method for managing rules for designing a computer program, comprising:

accessing a plurality of military theory rules for a military theory;

accessing a plurality of legislated laws associated with the military theory;

identifying military theory rules required by the laws as a plurality of domain rules of the military theory, each domain rule being invariant;

defining a domain from the domain rules, the domain used to determine a problem space and a solution space;

identifying one or more requirements of the domain from one or more supplemental sources;

generating a model that establishes the requirements of the domain;

designating the other military theory rules as a plurality of business rules of the military theory, the business rules comprising a plurality of rules engagement, each business rule being variable; and

~~storing the rules of engagement;~~ and

providing a rule of engagement from the ~~stored~~ rules of engagement in response to a request for the business rule.

22. (Currently Amended) The method of Claim 21, further comprising:

customizing the provided rule of engagement;

associating the customized rule of engagement with ~~a procedure;~~ the model; and

generating a code corresponding to the ~~procedure~~ model in order to design a computer program.

23. (Currently Amended) The method of Claim 21, further comprising:

associating the domain rules with ~~a procedure;~~ the model; and

generating a code corresponding to the ~~procedure~~ model in order to design a computer program.

24. (Original) The method of Claim 21, further comprising:
allocating the domain rules and the business rules to a plurality of use cases;
realizing the use cases; and
assessing the domain rules and the business rules in accordance with the realization.

25. (Currently Amended) A system for managing rules for designing a computer program, comprising:

a database operable to:

store a plurality of military theory rules for a military theory; and

store a plurality of legislated laws associated with the military theory; and

a server coupled to the database and operable to:

identify military theory rules required by the laws as a plurality of domain rules of the military theory, each domain rule being invariant;

define a domain from the domain rules, the domain used to determine a problem space and a solution space;

identify one or more requirements of the domain from one or more supplemental sources;

generate a model that establishes the requirements of the domain;

designate the other military theory rules as a plurality of business rules of the military theory, the business rules comprising a plurality of rules engagement, each business rule being variable; and

~~store the rules of engagement; and~~

provide a rule of engagement from the ~~stored~~ rules of engagement in response to a request for the business rule.

26. (Currently Amended) The system of Claim 25, wherein the server is further operable to:

customize the provided rule of engagement;

associate the customized rule of engagement with ~~a procedure;~~ the model; and

generate a code corresponding to the ~~procedure~~ model in order to design a computer program.

27. (Currently Amended) The system of Claim 25, wherein the server is further operable to:

associate the domain rules with ~~a procedure~~; the model; and
generate a code corresponding to the ~~procedure~~ model in order to design a computer program.

28. (Original) The system of Claim 25, wherein the server is further operable to:
allocate the domain rules and the business rules to a plurality of use cases;
realize the use cases; and
assess the domain rules and the business rules in accordance with the realization.

29. (Currently Amended) Logic for managing rules for designing a computer program, the logic embodied in a computer-readable medium and when executed by a computer operable to:

access a plurality of military theory rules for a military theory;

access a plurality of legislated laws associated with the military theory;

identify military theory rules required by the laws as a plurality of domain rules of the military theory, each domain rule being invariant;

define a domain from the domain rules, the domain used to determine a problem space and a solution space;

identify one or more requirements of the domain from one or more supplemental sources;

generate a model that establishes the requirements of the domain;

designate the other military theory rules as a plurality of business rules of the military theory, the business rules comprising a plurality of rules engagement, each business rule being variable; and

~~store the rules of engagement;~~ and

provide a rule of engagement from the ~~stored~~ rules of engagement in response to a request for the business rule.

30. (Currently Amended) The logic of Claim 29, further operable to:

customize the provided rule of engagement;

associate the customized rule of engagement with ~~a procedure;~~ the model; and

generate a code corresponding to the ~~procedure~~ model in order to design a computer program.

31. (Currently Amended) The logic of Claim 29, further operable to:

associate the domain rules with ~~a procedure;~~ the model; and

generate a code corresponding to the ~~procedure~~ model in order to design a computer program.

32. (Original) The logic of Claim 29, further operable to:
allocate the domain rules and the business rules to a plurality of use cases;
realize the use cases; and
assess the domain rules and the business rules in accordance with the realization.

33. (Currently Amended) A system for managing rules for designing a computer program, comprising:

means for accessing a plurality of military theory rules for a military theory;

means for accessing a plurality of legislated laws associated with the military theory;

means for identifying military theory rules required by the laws as a plurality of domain rules of the military theory, each domain rule being invariant;

means for defining a domain from the domain rules, the domain used to determine a problem space and a solution space;

means for identifying one or more requirements of the domain from one or more supplemental sources;

means for generating a model that establishes the requirements of the domain;

means for designating the other military theory rules as a plurality of business rules of the military theory, the business rules comprising a plurality of rules engagement, each business rule being variable; and

~~means for storing the rules of engagement; and~~

means for providing a rule of engagement from the ~~stored~~ rules of engagement in response to a request for the business rule.

34. (Currently Amended) A method for managing rules for designing a computer program, comprising:

accessing a plurality of military theory rules for a military theory;

accessing a plurality of legislated laws associated with the military theory;

identifying military theory rules required by the laws as a plurality of domain rules of the military theory, each domain rule being invariant;

designating the other military theory rules as a plurality of business rules of the military theory, the business rules comprising a plurality of rules engagement, each business rule being variable;

defining a domain from the domain rules, the domain used to determine a problem space and a solution space;

identifying one or more requirements of the domain from one or more supplemental sources;

generating a model that establishes the requirements of the domain;

allocating the domain rules and the business rules to a plurality of use cases;

realizing the use cases;

assessing the domain rules and the business rules in accordance with the realization;

storing the rules of engagement;

providing a rule of engagement from the stored rules of engagement in response to a request for the business rule;

customizing the provided rule of engagement;

associating the customized rule of engagement with a ~~procedure~~; the model;

associating the domain rules with the ~~procedure~~; model; and

generating a code corresponding to the ~~procedure~~ model in order to design a computer program.

35. (Canceled)

36. (Canceled)

37. (Canceled)

38. (Canceled)

39. (Canceled)

40. (Canceled)

41. (Canceled)

42. (Canceled)

43. (Canceled)

44. (Canceled)

45. (Canceled)

46. (Canceled)

47. (Canceled)

48. (Canceled)

49. (Canceled)

50. (Canceled)

51. (Canceled)

52. (Canceled)

53. (Canceled)

54. (Canceled)